

# Revisión y Clasificación de Protocolos para Redes de Tecnología ATM

José Luis González-Sánchez (jlgs@unex.es)<sup>(1)</sup> y Jordi Domingo-Pascual (jordid@ac.upc.es)<sup>(2)</sup>

<sup>(1)</sup> Universidad de Extremadura. Departamento de Informática. Área Arquitectura y Tecnología de Computadores.

<sup>(2)</sup> Universidad Politécnica de Catalunya. Departament d'Arquitectura de Computadors.

## Resumen

La tecnología *ATM* (*Asynchronous Transfer Mode*) continua evolucionando y siendo fuente de interesantes y novedosas propuestas. El desarrollo de avanzados protocolos de comunicaciones es uno de los campos de investigación más activo, con la aspiración de ofrecer el adecuado soporte a las nuevas aplicaciones adaptadas a las clases de servicio nativas *ATM*. En este contexto son aspectos clave los relativos a los protocolos nativos *ATM*, así como las características *multicast*, la *escalabilidad* y la *fiabilidad*. Se profundiza en el concepto de *protocolo nativo ATM* y son introducidos los protocolos más importantes que satisfacen este importante requerimiento (*N<sup>3</sup>*, *CONGRESS* y *kStack*, entre otros). Es importante destacar también los protocolos de transporte pensados específicamente para la tecnología *ATM*. La característica *multicast* (*multipoint-to-multipoint*) es una de las que más esfuerzo está costando garantizar a *ATM*, pero existen ya propuestas que permiten la comunicación fiable a elevados anchos de banda y entre múltiples emisores y receptores (*SMART*, *MCMP* o *MWAX*). El artículo concluye con las investigaciones más novedosas en torno a los protocolos *ATM* como las iniciativas que proponen redes *ATM* programables o activas (*active networks*) usando agentes móviles.

## 1 Introducción y motivaciones

La actual demanda de aplicaciones relacionadas con información multimedia, como son la video-conferencia, audio-conferencia, video bajo demanda (*VoD*) o sistemas colaborativos (pizarras compartidas, teletrabajo, telemedicina, etc.) y su coexistencia con aplicaciones más clásicas (bases de datos, transferencias de ficheros, *WWW*, etc.), requiere tecnologías de comunicaciones capaces de ofrecer elevadas prestaciones. Estas elevadas prestaciones están directamente relacionadas con la calidad de servicio (*QoS*) y concretamente con conceptos claramente parametrizables como el ancho de banda y la velocidad de transmisión (*throughput*), el retardo de las transferencias (*delay*); la variabilidad en el retardo (*jitter*); la fiabilidad (*reliability*) de las transmisiones; las características de multidifusión a grupos dispersos de usuarios (*multicast*) y la posibilidad de gestionar múltiples clases de servicio o flujos de información en redes *multiclass*.

Para que las nuevas tecnologías en comunicaciones puedan ofrecer estas características es necesario revisar, potenciar y ampliar las actuales arquitecturas, servicios y protocolos de comunicaciones. En los últimos dos o tres años, las investigaciones en el campo de *ATM* están dando lugar a importantes propuestas cuyo principal objetivo es ofrecer a las aplicaciones demandadas actualmente algunas o todas las características citadas anteriormente.

Presentamos en este trabajo los conceptos y propuestas más novedosas en el contexto de *ATM* para ayudar al lector interesado en el dinámico, complejo y sofisticado campo de los protocolos de comunicaciones. Realizamos la revisión de algunos de los más importantes conceptos, técnicas, ideas y mecanismos en protocolos de altas prestaciones para redes de tecnología *ATM*. El principal objetivo de este documento es ofrecer una actualizada visión general, no extensiva ni profunda, de los protocolos propuestos y en desarrollo para dotar a las diversas clases de servicio *ATM* de las citadas características de calidad de servicio que aporten las potenciales elevadas prestaciones que la tecnología *ATM* es capaz de ofrecer.

El resto del documento está estructurado de la siguiente forma. La sección 2 revisa el concepto de protocolo nativo centrado en el contexto de las redes *ATM*. La sección 3 describe las propuestas más interesantes actualmente en materia de protocolos de transporte. El apartado 4 comenta las transferencias *multicast* como un importante objetivo para una tecnología orientada a la conexión como es *ATM*. Concluimos en la sección 5 con los campos de investigación abiertos recientemente como son las redes activas y los protocolos *booster*.

## 2 Protocolos nativos ATM

Las aplicaciones nativas ATM están específicamente pensadas para usar la tecnología ATM y para explotar al máximo sus especiales características. Los protocolos nativos se encargan, por tanto, de ofrecer esas características intrínsecas de las redes de tecnología ATM (soporte de QoS, señalización, direccionamiento, etc.) a las aplicaciones nativas ATM (VoD, pizarras compartidas, video-conferencia...). No obstante, existen también activas investigaciones para conseguir soportar sobre redes ATM aplicaciones no nativas ATM desarrolladas para otras tecnologías (IP, Frame Relay, SMDS...).

En [1], el término *native ATM services* define servicios ATM específicos disponibles para el software y hardware residentes en dispositivos de usuario UNI ATM. Por consiguiente, el programador de aplicaciones dispone de nuevos servicios entre los que se pueden destacar los siguientes:

- Transferencias de datos (fiabes o no) usando la capa ATM y varias capas de adaptación (AALs).
- Disponibilidad de circuitos virtuales conmutados (SVCs) y circuitos virtuales permanentes (PVCs).
- Consideraciones relativas a la gestión de tráfico (clases de servicio, garantías de QoS, etc.).
- Posibilidad de distribución de conexiones y de participación local en la administración de la red (protocolos ILMI y OAM).

El propósito de los servicios nativos ATM es ofrecer el acceso a las clases de servicio o a las características de QoS en redes ATM. Estos servicios nativos también ofrecen soporte a un amplio y heterogéneo rango de flujos con diversas propiedades y requerimientos recomendados en [1].

Los protocolos de transferencia nativos ATM gestionan la señalización UNI para establecer los SVCs, configurar PVCs y mapear los perfiles de QoS en la correspondiente clase de servicio. Los protocolos nativos también realizan funciones clásicas como las de transporte, mecanismos de control de errores, transferencia de datos, y controles de flujo y de congestión.

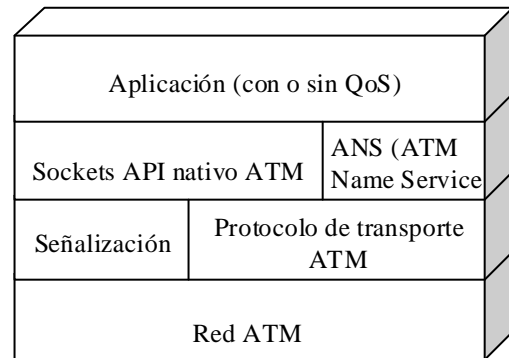
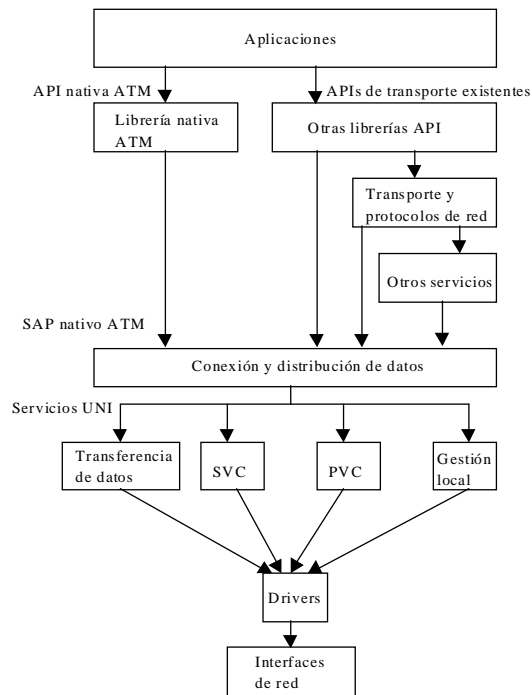
En la referencia [1] se especifica la definición semántica de los servicios y consideramos que es útil contrastar esta semántica con las redes ATM actuales que usan TCP como capa de transporte e IP-over-ATM como capa de red, planteamiento, por otro lado, poco adecuado [2] por las siguientes razones:

- Las redes IP no garantizan la QoS extremo-extremo ofrecida por las redes ATM a circuitos individuales. IP multiplexa múltiples conexiones de transporte con distintos requerimientos de QoS en VCs simples.
- TCP no soporta células RM (Resource Management) ABR y por consiguiente no puede usar directamente las garantías de QoS ofrecidas por la red.
- ATM Adaptation Layer 5 (AAL5) realiza labores de checksum para detectar corrupción de datos. TCP también realiza estas labores (redundantes con AAL5) costosas en overhead (cada byte de un paquete debe ser chequeado).
- TCP/IP son la representación de un grupo de protocolos bastante más antiguos que ATM y que ya han experimentado determinados arreglos y evoluciones lo mismo que las aplicaciones que los emplean, lo que acaba dando, en algunos casos, inadecuados comportamientos.

Estos y otros problemas son eliminados usando pilas de protocolos en modo nativo ATM que son revisados en este apartado. La Fig. 1 muestra el modelo de referencia para servicios nativos ATM [1]. Además, las siguientes razones [2] justifican el desarrollo de pilas de protocolos nativos ATM:

- En la actualidad existen muchas aplicaciones pensadas para explotar avanzados servicios usando tecnología ATM y también existen antiguas y no nativas aplicaciones. Este escenario implica cambiar las aplicaciones o proponer nuevas pilas de protocolos nativos ATM.
- La encapsulación consecutiva de paquetes genera problemas de overhead y funciones redundantes como se ha argumentado anteriormente.
- La limitación de recursos en los sistemas finales es otra importante motivación para usar pilas de protocolos nativos y ligeros.
- La QoS ofrecida por el modo nativo es aprovechada por los usuarios para demandar recursos a los proveedores de servicios en redes privadas. Los proveedores de servicios públicos disfrutan también de estas ventajas.
- ATM, RDSI y la telefonía ofrecen un esquema de direccionamiento universal basado en NSAP/E.164 el cual es capaz de enrutar tráfico de forma nativa. Por tanto, aunque ATM dispone de protocolos nativos con direccionamiento intrínscico estructurado y jerárquico, éste no es aprovechado por las aplicaciones que están basadas en IP. El esquema de direccionamiento ATM es una de las principales dificultades en los protocolos propuestos como nativos [2].

ATM Forum ha definido las especificaciones y también existen importantes investigaciones en torno a los protocolos nativos ATM. En esta sección se muestran las propuestas más importantes y actuales en materia de protocolos nativos ATM [2, 7-12] que, a su vez, están sirviendo de base para nuevas investigaciones.



**Fig. 1** Modelo de referencia para servicios nativos ATM [1] **Fig. 2** Visión de alto nivel de la pila de protocolos  $N^3$  [2]

[8] presentan el diseño, implementación y comportamiento de una pila en modo nativo *ATM* y contrasta la semántica de su capa de transporte con *TCP*. Este trabajo es diferente a *IP-over-ATM*, y justifica el uso de la pila nativa *ATM* para solventar automáticamente los siguientes problemas:

- *IP-over-ATM* no ofrece garantía de *QoS* pues sus aplicaciones sólo “ven” la interfaz *IP*.
- El núcleo de los sistemas operativos y los sistemas finales son sobrecargados con considerable complejidad pues el subsistema *IP-over-ATM* debe encargarse de las peticiones de la señalización.
- *IP-over-ATM* debe emular el *routing IP* sobre conexiones *punto-a-punto* de la red *ATM* lo que supone pagar un elevado precio en prestaciones.

La capa de transporte propuesta en [8] ha sido construida para minimizar el *overhead* y sacar ventaja de *AAL5*. Ofrece, entre otras características, la entrega fiable y no fiable de datos con control de flujo. Este protocolo de transporte es ampliado en la sección 5.

[2] presenta  $N^3$  (*Native Non-broadcasting medium access Networking*) un protocolo de transporte ligero y nativo *ATM*. La pila  $N^3$  se ha diseñado para ofrecer avanzados servicios multimedia a comunidades residenciales. El documento describe una arquitectura capaz de ofrecer aplicaciones nativas *ATM*. La pila de protocolo de transporte  $N^3$  se basa en implementaciones previas [8] y, además, aporta otras importantes novedades. Los principales componentes de  $N^3$  incluyen una *API sockets ATM* nativa, un protocolo de transporte *ATM* y un servicio de nombres *ATM* (**Fig. 2**). El protocolo de transporte *ATM* propuesto en [2] ofrece soporte para tres diferentes tipos de servicio: entrega garantizada (mecanismos de retransmisión), velocidad garantizada (mecanismo *leaky bucket*) y servicio *best-effort*. Otro objetivo principal de la pila *ATM* es su compatibilidad con aplicaciones tradicionales sin necesidad de recompilaciones.

[9] presenta una arquitectura de servicio *ATM* en modo nativo capaz de ofrecer a las aplicaciones nativas *ATM* acceso completo a las diversas clases de servicio *ATM*. Los elementos de la arquitectura propuesta se responsabilizan de las transferencias eficientes de datos sobre *ATM*, del control de errores *extremo-extremo*, del control de flujo y congestión de la transferencia de *datagramas* y de la multiplexación de *VCs*. La referencia [9] introduce los componentes de la arquitectura, sus funcionalidades y capacidades. Los mayores esfuerzos del protocolo se centran en maximizar la efectividad del rendimiento *extremo-extremo* en canales de datos que usan la clase de servicio *UBR*. La **Fig. 3** presenta los elementos de *Native Mode Service Architecture* donde el *Flow Management* es el componente más importante. El *Flow Management* se responsabiliza de manipular los flujos de datos desde y hasta la red vía la interfaz *AAL*. La segmentación, el reensamblado y el control de errores es también realizada por esta entidad. Para las clases de servicio *CBR*, *VBR* y *ABR* se emplea un sencillo esquema de control llamado *Back-Pressure Flow*. Para servicios *UBR* se emplea un control de congestión y de flujo *extremo-extremo* más complejo.

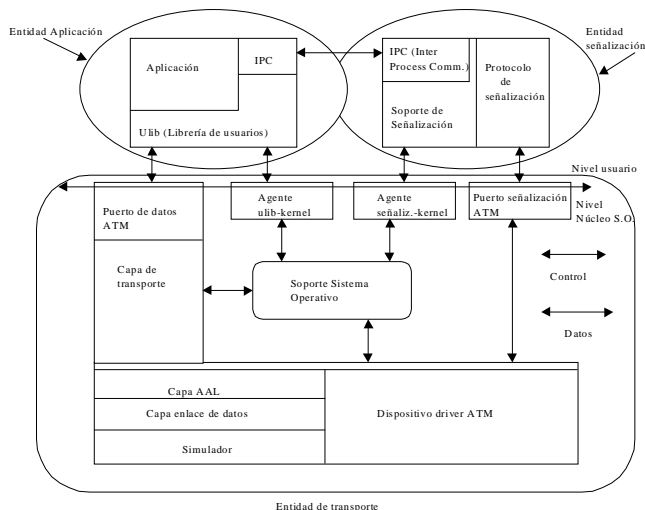


Fig. 3 Arquitectura de servicio nativo ATM [9]

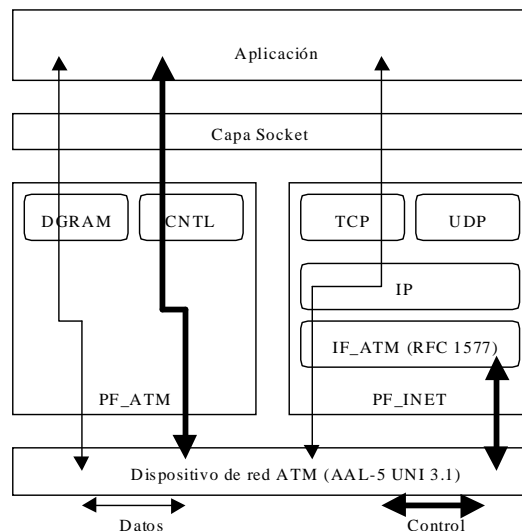


Fig. 4 Descripción arquitectura de protocolo [7]

[7] describe la semántica de una pila de protocolos y explora una nueva arquitectura de protocolo adaptada a la tecnología ATM y a las aplicaciones multimedia. El diseño está basado en tres principios básicos: separación de flujos de control y de datos; minimización del *overhead* y de la duplicidad de funciones; y acceso de las aplicaciones al nivel ATM con garantías de *QoS*.

La idea es mezclar el soporte nativo ATM en la estructura existente del protocolo (Fig. 4) que muestra dos caminos separados en el protocolo: la familia nativa ATM y la familia del protocolo IP. Las aplicaciones que tienen acceso transparente a la red ATM usan la familia del protocolo PF\_INET. El mapeo de IP en ATM es gestionado por la interfaz de red ATM (IF\_ATM) usando el protocolo IP-over-ATM.

La interfaz *Native ATM* es constituida por la familia de protocolos PF\_ATM que es directamente soportada encima del dispositivo de red ATM sobrepasando la capa interfaz de red. El módulo CNTL abre una conexión de señalización con el dispositivo ATM y establece una gestión de las llamadas de mensajes de configuración.

PF\_ATM separa flujos de datos y de control para aliviar el límite de comportamiento en las comunicaciones. Esto permite a los mecanismos de control de tráfico ser rápidos y sencillos, mientras los mecanismos de control pueden ser tan complicados como sea necesario. Esta separación permite también que los dispositivos puedan estar en los puntos finales de una conexión.

La interfaz PF\_ATM da a las aplicaciones acceso directo a la capa de enlace ATM y extiende las garantías de *QoS* a los puntos extremos de la comunicación.

Un segundo prototipo [7] es diseñado e implementado con dos objetivos principales en la optimización de la pila nativa ATM:

- Optimizar los caminos de datos entre los adaptadores ATM aprovechando la separación de flujos de control y de datos y la capacidad de gestión de datos específicos de las conexiones de la pila ATM.
- Optimizar el procesamiento de *overheads* usando una pila de protocolos nativo ATM en lugar de UDP/IP.

[11] introduce CONGRESS (*CON*nection oriented *GR*oup-address *RES*olution *S*ervice), otro eficiente protocolo nativo ATM para la resolución y gestión de direcciones de grupos *multicast* en una red ATM. El servicio CONGRESS resuelve direcciones de grupo *multicast* y mantiene los miembros pertenecientes a esos grupos para uso de las aplicaciones. CONGRESS ofrece escalabilidad con su diseño basado en los dos siguientes principios:

- Diseño jerárquico: los servicios del protocolo son ofrecidos a las aplicaciones por múltiples servidores organizados jerárquicamente.
- No inundación: se evita la inundación de la WAN en cada cambio de grupos *multicast*.

La referencia [12] presenta *kStack*, una nueva capa de transporte nativa ATM en el espacio de usuario con soporte de *QoS*. Esta implementación sobre *Unix* y *Windows NT* está basada y es compatible con los trabajos originales de Ahuja, Keshav y Saran [8] *Native-Mode ATM Stack* comentados anteriormente. El protocolo *kStack* es similar al original, pero se ha modificado sustancialmente en aspectos como su implementación en el espacio de usuario, se ha ampliado implementando una capa de transporte con *QoS* y se ha añadido un módulo que monitoria la *QoS extremo-a-extremo*.

### 3 Protocolos de transporte para redes ATM

En los dos o tres últimos años ha habido numerosos e interesante intentos por diseñar protocolos de transporte. La referencia [10] presenta un completo y, ya clásico, resumen de las propuestas más interesantes en materia de protocolos de transporte para redes de alta velocidad (mayoritariamente *IP* en el artículo). Además, la referencia [13] ofrece una interesante visión de las características más importantes en los protocolos de elevada velocidad. Son estudiadas también diversas arquitecturas de protocolos y varias técnicas de implementación. Los protocolos de transporte de elevada velocidad son fuente de activas investigaciones y están en constante evolución desde hace más de dos décadas, lo que impide ser exhaustivo en este resumen: no citamos todos los protocolos, ni presentamos todos los conceptos ni podemos analizar todas las soluciones.

Uno de los componentes del ámbito de las comunicaciones que ha recibido mayor atención es la capa de transporte, la cuarta capa del *OSI-RM* de los protocolos de comunicaciones. *TCP* e *ISO TP4* son los dos más populares protocolos de transporte. Pero centrándonos más concretamente en el ámbito de *ATM*, la literatura presenta varios protocolos de transporte y arquitecturas para redes de alta velocidad revisadas resumidamente a continuación.

[7] ofrece una excelente y didáctica arquitectura de pila de protocolos para aplicaciones multimedia en modo nativo *ATM*. Esta arquitectura de protocolo ha sido ya descrita brevemente en la sección anterior.

[8] presentan el diseño, implementación y comportamiento de un protocolo situado en la capa de transporte *ATM*. Esta es una interesante propuesta en la que se puede destacar que la pila *ATM* está formada por tres entidades principales: aplicación, señalización y transporte.

La siguiente **Tabla 1** muestra un conjunto de nueve básicos servicios ortogonales que pueden ser combinados para obtener los requerimientos de determinadas aplicaciones. Actualmente, la capa de transporte referenciada en [8] soporta tres clases de servicio o combinación de servicios. La marca **X** indica el servicio básico soportado en cada clase de servicio general.

Servicios	Clases de servicio		
	Servicio de comportamiento garantizado	Servicio fiable	Servicio <i>Best effort</i>
- Transferencia Simplex de datos	X	X	X
- Control de errores	-	deteccion de errores, <i>timeouts</i> , retransmisiones	No existente (ofrecido por AAL5)
- Bucle abierto	No existente	-	-
- Control de flujo realimentado	-	X	No existente
- Tamaño de mensaje ilimitado	X	X	X
- Elección de blocking	X	X	X
- Aplicación Non-blocking	X	X	X
- Elección de byte stream	X	X	X
- Semántica transferencia de mensajes	X	X	X
- Garantías de QoS	requerimientos de ancho de banda	No soportado	No soportado
- Reserva de recursos	X	No existente	No existente
- Transferencias Multicast	X	No soportado	Para servicios no fiables

**Tabla 1 Servicios ortogonales y clases de servicio**

[15] resuelve el problema de soportar *HPDC* (*High Performance Distributed Computing*) sobre redes *ATM*. Los autores sugieren modificaciones en los mecanismos de recuperación de pérdidas del protocolo estándar *SSCOP* [14] y demuestran que el resultado ofrece baja latencia, eficiente recuperación de células perdidas y es tan robusto como el estándar *SSCOP*.

### 4 Protocolos multipoint

El crecimiento de las redes *ATM* viene motivado, en parte, por la demanda de servicios multimedia para grupos dispersos de usuarios. El tráfico *multicast* tiene características particulares descritas para *ATM* en *UNI 4.0* [6] y anteriores [2,5]. La distribución de información *punto-a-multipunto* (*uno-a-muchos*) o *multipunto-a-multipunto* (*muchos-a-muchos*) es un objetivo básico propuesto por varios protocolos y arquitecturas *ATM* que ofrecen el soporte multimedia y/o *multicast* como audio-conferencia, video-conferencia, trabajos colaborativos o *VoD*.

*ATM* es aún una tecnología emergente diseñada para ser usada por aplicaciones de datos, audio y video, lo que requiere un buen comportamiento de las transferencias *unicast* y *multicast*. *User Network Interface (UNI 3.0)* para *ATM* define conexiones *punto-a-multipunto*, y las conexiones *multipunto-a-multipunto* sólo pueden ser obtenidas de las dos siguientes formas:

- El primer esquema consiste en configurar N conexiones *punto-a-multipunto* para conseguir conectar todos los nodos en una topología completamente mallada *todos-con-todos*. Aunque esta topología ofrece conexiones *multipunto-a-multipunto*, hay que destacar que no escala bien cuando el número de participantes es elevado.
- Una alternativa al anterior esquema es el uso de un servidor que actúa a modo de raíz en el árbol *multipunto*. Este método sólo requiere un nodo raíz para almacenar información, pero la desventaja de este método son las potenciales congestiones en el servidor cuando debe encargarse de envíos y retransmisiones de las conexiones *multipunto-a-multipunto*.

Para solventar las limitaciones de *UNI 3.0* y *UNI 3.1* [5] que soportan conexiones *uno-a-muchos*, pero no directamente (nativamente) conexiones *muchos-a-muchos*, y ofrecer a *ATM* verdadero servicio *multicast*, *ATM Forum*, *ITU-T* e *IETF* han realizado varias propuestas al actual mecanismo de señalización *ATM* (*UNI 3.1*, *UNI 4.0*), [4,5,6].

Comenzamos destacando la referencia [16] como un reciente trabajo que revisa y compara los protocolos de transporte *multicast* más importantes en *Internet* como *MTP-2*, *XTP*, *RTP*, *SRM*, *RAMP*, *RMTP*, *MFTP*, *STORM*, etc.. *IETF* e *IRTF* (como *ITU-T* y *ATM Forum*) también impulsan una importante actividad en este campo. La referencia [16] revisa los más representativos protocolos *multicast* y los clasifica de acuerdo a la taxonomía de varias características (propagación de datos, mecanismos de fiabilidad, retransmisiones, control de congestión y de flujo, gestión de grupos *multicast*, etc.). En *Internet* los mecanismos efectivos de control de congestión son una de las prioridades en las investigaciones de las transferencias *multicast* fiables. Los mecanismos de seguridad y las técnicas escalables de recuperación de errores son algunos de los aspectos actualmente en estudio en el campo de los protocolos de transporte *multicast*.

Hasta este punto hemos revisado algunos protocolos de transporte *ATM* y hemos citado sus más importantes características. En esta sección comentaremos los problemas asociados a las transferencias *multicast uno-a-muchos* o *muchos-a-muchos*. Actualmente no existen excesivas propuestas en este importante faceta para *ATM*, pero vamos a resumir algunas de las más interesantes en los siguientes párrafos.

*SMART* (*shared many-to-many ATM reservations*) [3] es un protocolo para controlar un árbol *ATM multicast* compartido soportando comunicaciones *muchos-a-muchos* (*many-to-many*). Esta propuesta tiene importantes características como que: reside completamente en la capa *ATM* y no requiere ningún servidor; soporta uno o varios *VCCs* (y también *VPCs*) cuyo número es libremente configurado y es independiente del número de puntos finales; usa el concepto de bloques de datos como en la clase de servicio *ABT* y también permite *VCCs* de las clases *CBR*, *VBR* o *UBR*; el protocolo garantiza que no existen puntos de interrelación en los *VCC* del árbol; son respetadas las garantías del contrato de tráfico asociado con los *VCCs*, etc. *SMART* puede ser entendido como un protocolo completamente distribuido para coordinar la distribución de *VPIs/VCI*s.

Para solventar las conocidas dificultades debidas al soporte y uso de *muchos-a-muchos VCCs*, *SMART* usa el mecanismo de *Cell Interleaving* (sobre un *VCC muchos-a-muchos*, las células de datos desde diferentes fuentes pueden llegar intercaladas a un destinatario) y también *Demand Sharing* (los recursos asignados a *conexiones muchos-a-muchos* son dinámicamente compartidas entre todas las potenciales fuentes).

El artículo [3] describe el protocolo *SMART* formal e informalmente, y propone completas pruebas de corrección y un detallado análisis de comportamiento para estudios futuros. También son sugeridas otras investigaciones como son: ofrecer justicia en los accesos a los árboles *multicast*; investigar las células *RM* periódicamente para aliviar las congestiones en la red o disminuir el tiempo de acceso del usuario a los árboles de distribución *multicast*; análisis de las células *RM* dentro de cada *VCC* o fuera enviando todas las células *RM* en un *VCC* dedicado.

En [17] se presenta *MWAX* un algoritmo dinámico y escalable para *routing multicast* en el marco *PNNI* de redes *ATM*. Los autores han identificado el problema para conseguir la escalabilidad con protocolos *multipunto-a-multipunto* y se proponen soluciones para este problema. El artículo describe un esquema jerárquico basado en *CBT* para incorporar *routing multipunto-a-multipunto* en *PNNI*. En el algoritmo los nodos *core* actúan como participantes pasivos para eliminar la dependencia en la selección de estos nodos. Con un mecanismo de *backup* se consigue un algoritmo tolerante a fallos en los nodos *core*, lo cual puede ser fácilmente extendido para incorporar *QoS* en el *routing multicast*. El protocolo-algoritmo *MWAS* es recursivo, esto es, el mismo protocolo es ejecutado en cada nivel de la jerarquía.

*SEAM* (*Scalable and Efficient ATM Multicast*) [18] propone una arquitectura escalable, eficiente y *multicast multipunto-a-multipunto* para redes *ATM* que usa un sólo *VC* para un grupo *multicast* de múltiples emisores y receptores y todo ello sin realizar cambios en la capa *AAL5* de *ATM*. Esta propuesta permite a los grupos *multicast* aprovechar el soporte de *QoS* y la escalabilidad del ancho de banda. También realiza aportaciones para conseguir soportar *IP multicast* sobre redes *ATM* extensas.

*SEAM* usa un sólo árbol de distribución compartido para todos los emisores y receptores. Cada grupo *multicast* tiene un *core* asociado, el cual se usa como punto focal para todos los mensajes de señalización del grupo. Este trabajo deja abiertas investigaciones referentes a la gestión de tráfico y a la entrega fiable de tráficos *multicasting*.

Concluimos esta sección destacando *MCMP* (*Multiparty Conference Management Protocol*) [19] que, sin estar pensado específicamente para *ATM*, es un protocolo de nivel sesión/transporte distribuido *extremo-extremo* y desarrollado para gestión de grupos de aplicaciones de conferencia. *MCMP* es un conjunto de algoritmos de control distribuido para configuración de conferencias *multipunto* y gestión de miembros de grupos de usuarios. Conceptualmente, *MCMP* reside en el nivel de sesión en el que se establece la infraestructura para activar la transferencia de información entre los participantes en la conferencia. Pero funcionalmente, el protocolo acompaña los niveles de sesión y de transporte pues utiliza directamente servicios del nivel de red. Son destacables las condiciones de corrección (conectividad, validación, unicidad, consistencia y terminación) que deben ser satisfechas una vez que la conferencia ha sido configurada por el algoritmo de configuración de *MCMP*. El artículo muestra exhaustivas pruebas de corrección para uno de los algoritmos, y también describe la especificación y verificación del protocolo.

## 5 Nuevos campos de investigación

En todas las redes existen gran variedad de elementos hardware (*switchs, routers, bridges, routers, hubs, ETDs*, etc.) que realizan muy diversas funciones (conmutación, *routing*, puenteo, controles de congestión y de flujo, garantía de *QoS*, ejecución de aplicaciones, etc.). En la actualidad la red es mayormente un canal de comunicación para transferir paquetes entre equipos finales (ayudada por los elementos hardware antes citados). Pero también se están realizando importantes esfuerzos para equipar a los elementos hardware con elevadas prestaciones aportadas por diversas técnicas software. Esto dota a la red de características activas (*active networks*) en el sentido de que los elementos hardware que la componen computan, modifican u operan los contenidos de los paquetes y también serán capaces de transferir o propagar código. Por consiguiente, una red activa es una red programable.

[20] es una excelente revisión de este novedoso campo de investigación que discute dos planteamientos en la realización de redes activas, la idea del conmutador programable y la de la cápsula. Una red es activa si en sus árboles de distribución *multicast* existen nodos activos con capacidad para ejecutar programas y/o capaces de implementar mecanismos de propagación de código. Algunas de las ventajas de los protocolos activos son conseguidas instalando nodos activos en puntos estratégicos de la red.

Otro nuevo concepto es presentado en [21] como una metodología para el diseño de protocolos. Los *protocol boosters* son una nueva contribución a las redes activas o programables. Una ventaja de los *boosters* es que pueden ser fácilmente “inyectados” en los sistemas actuales sin provocar cambios en la infraestructura de red. Por otro lado, [22] propone el concepto de agente de comunicaciones para servicios de comunicaciones multimedia en redes de área extensa. Este papel introduce una arquitectura software orientada a agente y propone el concepto de agente de comunicación para servicio de comunicación multimedia. Los servicios multimedia son expresados como agentes.

Los conceptos como redes activas, protocolos *boosters* o agentes software han sido propuestos y desarrollados para redes *IP*, sin embargo, la actividad empieza a notarse en las redes *ATM*, y la referencia [23] es una de esas recientes investigaciones. Este artículo muestra agentes software móviles usados para implementar operaciones robustas y funciones de mantenimiento en redes *ATM*. Los agentes desempeñan un rol similar al de las células *OAM* en *ATM* estándar, pues son transmitidos entre entidades de control a intervalos regulares usando recursos predefinidos. La diferencia entre los agentes móviles y las células *OAM* reside en que éstos pueden contener código.

## Referencias

- [1] \_\_\_\_ “Native ATM Service: Semantic *Description Version 1*,” *ATM Forum Technical Committee*, ATM Forum Document af-saa-0048.000, (Feb. 1996).
- [2] T. Zahariadis, J. Sanchez-P, C. Georgopoulos, V. Nellas, T. Arvanitis, D. Economou, G. Stassinopoulos, “Native ATM Protocol Stack for Internet Applications in Residential Broadband Networks,” *Multimedia Applications Services and Techniques ECMAST’98*, Springer, (May 1998).
- [3] E. Gauthier, J. Le Boudec and P. Oechslin, “SMART: A many-to-many Multicast protocol for ATM,” *IEEE Journal on Selected Areas in Communications*. Vol. N° 3 (April 1997).
- [4] W. D. Zhong, K. Yukimatsu, “Design requirements and architectures for multicast ATM switching,” *IEICE Trans. Com.*, Vol E77-B, pp. 1420-1428, (Nov. 1994).
- [5] \_\_\_\_ “ATM user-network interface version 3.1 specification”, *ATM Forum*, (1994).
- [6] \_\_\_\_ “Traffic Management Specification Version 4.0,” *ATM Forum Technical Committee*, ATM Forum Document af-tm-0056.000, (April 1996).
- [7] D. Kandlur, D. Saha and W. Willebeck, “Protocol Architecture for multimedia Application over ATM Networks,” *IEEE Journal Selected and Communications Vol. 14*, N° 7, pp. 1.349-1.359, (Sep. 1996).
- [8] R. Ahuja, S. Keshav and H. Saran, “Design, Implementation, and Performance Measurement of a Native-Mode

- ATM Transport Layer (Extended Version)," *IEEE/ACM Transactions on Networking*, Vol 4, N° 4, (Aug. 1996).
- [9] R. Karabek, "A Native ATM Protocol Architecture Design and Performance Evaluation," *IEEE Proceedings 22<sup>nd</sup> Annual Conference on Local Computer Networks*, pp. 204-210, (1997).
- [10] D. C. Feldmeier "A framework of architectural concepts for high-speed communications systems," *IEEE J. Select. Areas Commun.*, Vol. 11, N° 4, (May 1993).
- [11] T. Anker, D. Breitgand, D. Dolev, Z. Levy, "Congress: CONnection-oriented Group-address RESolution Service," *Proceeding of SPIE'97 vol 3233 on Broadband Networking Technologies*, Nov. (1997).
- [12] *kStack* <http://comet.columbia.edu/software/kStack>
- [13] T. La Porta and M. Schwartz, "Architectures, Features, and Implementation of High-Speed Transport Protocols," *IEEE Network Magazine*, pp. 14-22, (May. 1991).
- [14] \_\_\_\_ "B-ISDN, ATM Adaptation Layer Service. Specific Connection Oriented Protocol (SSCOP) Q.2110," *ITU-T*, (10-Mar. 1994).
- [15] J. Solé-Pareta, J. Vila-Sallent, "Network-based parallel computing over ATM using improved SSCOP protocol," *Computer Communications* 19 pp. 915-926, (1996).
- [16] K. Obraczka, "Multicast Transport Protocols: A survey and Taxonomy," *IEEE Communi. Magazine*, (1998).
- [17] R. Venkateswaran, C.S. Raghavendra, X. Chen and V.P. Kumar, "A Scalable, Dynamic Multicast Routing Algorithm in ATM Networks," *IEEE*, pp. 1361-1365 (1.997).
- [18] Matthias Grossglauser and K.K. Ramakrishnan, "SEAM: Scalable and Efficient ATM Multicast," *IEEE* 1.997, pp. 867-875, (1.997).
- [19] M. Nguyen, and M. Schwartz, "MCMP: A Transport/session level Distributed Protocol for Desktop Conference Setup," *IEEE Journal Selected and communications*, Vol. 14 N° 7, pp. 1404-1421, (Sept. 1996)
- [20] D. Tennenhouse et al., "A Survey of Active Network Research," *IEEE Communications Magazine*, (1997).
- [21] David C. Feldmeier, Anthony J. McAuley. Jonathan M. Smith, Deborah S. Bakin, William S. Marcus and Thomas M. Releigh, "Protocol Boosters," *IEEE JSAC* Vol. 16, N° 3, pp. 437-443, (April 1.998).
- [22] R. Kishimoto, "Agent communication system for multimedia communication services," *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking the Next Generation., Proceedings IEEE Vol. 1*, pp. 10-17, (1996).
- [23] David A. Halls and Sean G. Rooney, "Controlling the Tempest: Adaptive Management in Advanced ATM Control Architecture," *IEEE JSAC* Vol. 16, N° 3, pp. 414-423, (April 1.998).